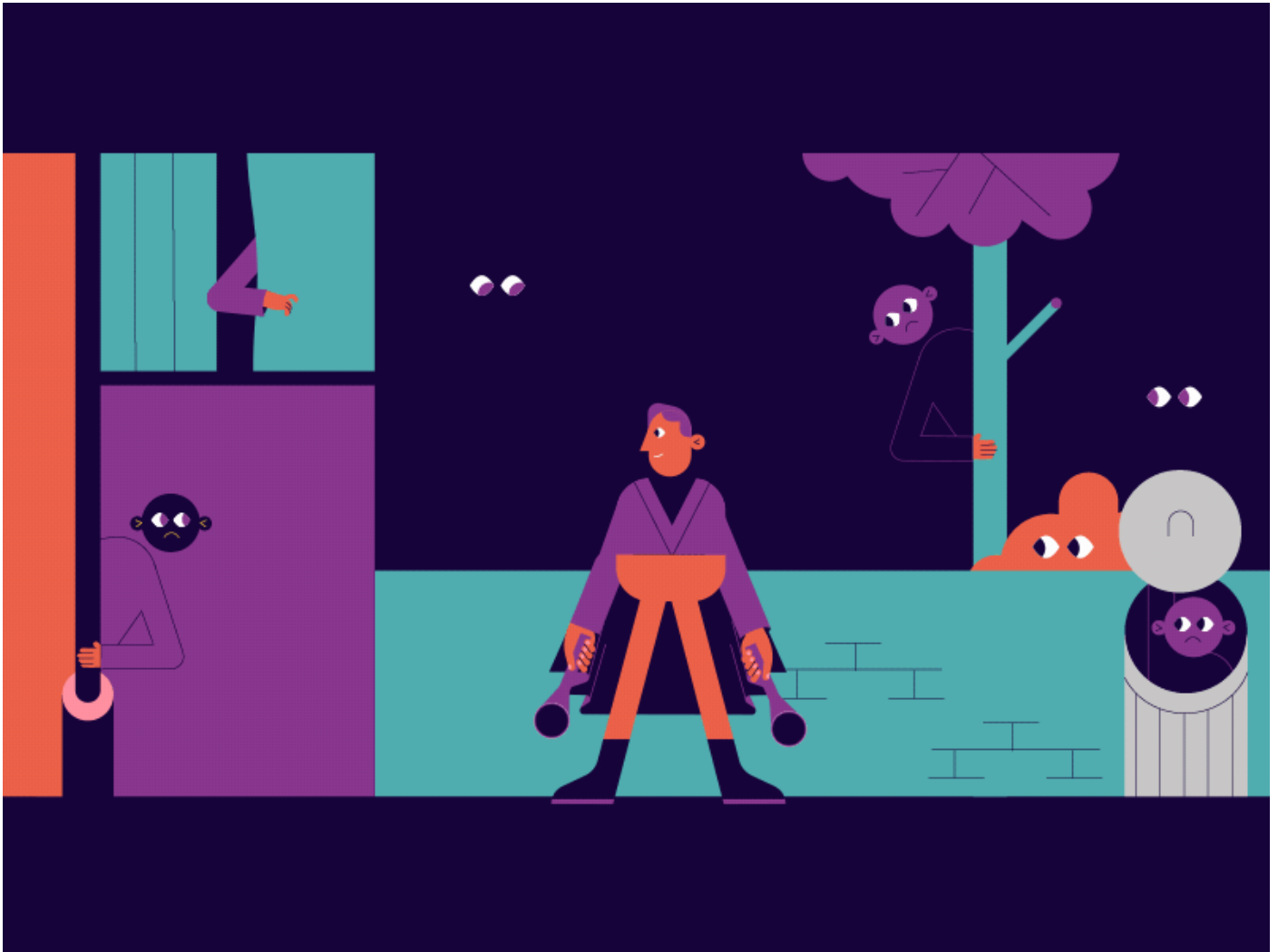


SECURITY RESEARCH

BYE JELLE ANDERS AND ART NOOIJEN



<https://cdn.dribbble.com/users/673873/screenshots/5310239/media/1d981ae5703fc1114024418fd84ce03d.gif>

Tania Yakunova

Inleiding.....	3
Informatie	3
Hoofdvraag.....	4
Deelvragen	4
Wat is het dot framework?.....	5
A Development oriented triangulation framework	5
The what of the research	5
THE WHY of the Research	5
The How of the research.....	6
Triangulation	6
DOT framework gebruik.....	7
Deelvragen beantwoord.....	8
Wat is een api key?.....	8
Hoe is een Apl key ontstaan?	9
Wanneer gebruik je een API key?	9
Wat is de beste manier om een api key te maken en te controleren?	10
Hoe sla je een api key op?.....	12
Bronvermelding	13

INLEIDING

In dit onderzoeksproject van Fontys Hogeschool, aangevoerd door Jelle Manders en Art Nooijen, onderzoeken we de beveiligingsaspecten van API-gebruik in applicaties. Centraal staat de vraag: "Hoe zorgt een API key ervoor dat een API beter beveiligd wordt?" Door een reeks gerichte deelvragen te onderzoeken, zoals de definitie, het ontstaan, en het juiste gebruik van API keys, alsook de creatie en opslag ervan, helpt dit ons om onze hoofdvraag te beantwoorden..

INFORMATIE

Jelle Manders en Art Nooijen, tweedejaarsstudenten aan de HBO-ICT-opleiding van Fontys Hogeschool met een specialisatie in software, voeren dit project uit onder begeleiding van Leon Bokhorst. Het doel van het onderzoek is inzicht verwerven in cybersecurity en ervaring opdoen met het DOT-framework.

HOOFDVRAAG

Onze hoofdvraag voor dit security research luidt als volgt:

“Hoe zorgt een API key ervoor dat een API beveiligd is?”

DEELVRAGEN

Om onze hoofdvraag duidelijk te kunnen beantwoorden hebben wij verschillende deelvragen opgesteld. Deze deelvragen voorzien ons van extra informatie om de hoofdvraag zo duidelijk mogelijk te beantwoorden.

Wat is een API key?

Hoe is een API key ontstaan?

Wanneer gebruik je een API key?

Wat is de beste manier om een API key te maken en te controleren?

Hoe sla je een API key op?

WAT IS HET DOT FRAMEWORK?

Het DOT-framework is een verzameling van onderzoeksmethodes die special ontworpen is voor praktijkgericht onderzoek binnen de ICT. Hier onder vallen strategieën zoals veldonderzoek bibliotheek onderzoek en labonderzoek meetbare resultaten, werkplaatsonderzoek en prototypes en showroomonderzoek. (Vogel, n.d.)^[1]

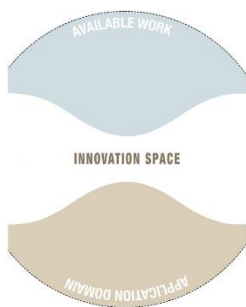
A DEVELOPMENT ORIENTED TRIANGULATION FRAMEWORK

“To help you in your ICT research, we advise the use of the Development Oriented Triangulation (DOT) framework ([1]). The DOT framework can help you to structure your research and to communicate about it. The Development Oriented Triangulation (DOT) framework consists of three levels:

- The "What" of your research (the domains)
- The "Why" of your research (the trade-offs)
- The "How" of your research (the strategies and methods)”

Quote from (Vogel, n.d.)^[1]

THE WHAT OF THE RESEARCH



https://ictresearchmethods.nl/app/immutable/assets/Three_Domains-adfec8cd.jpg

Hieronder vallen 3 onderdelen:

- Available work
- Innovative space
- Application domain

In het DOT-framework beschrijft het “wat” van ons onderzoek de domeinen waar het onderzoek zich in afspeelt. Het eerste domein is het toepassingsdomein gerelateerd aan de specifieke context waar het ICT-project zich plaatsvindt. Het tweede domein is het beschikbaar werk, omvat alle theorieën en modellen die wij kunnen gebruiken voor ons project. Het derde domein is het innovatieve domein waar ons advies/rapport tot stand komt gebaseerd op het onderzoek dat wij hebben gedaan.

THE WHY OF THE RESEARCH

Het “waarom” van ons onderzoek beschrijft het doel: het ontwikkelen van een passend product voor de stakeholders of het waarborgen van de kwaliteit. Het gaat om het combineren van data-gedreven onderzoek met het inspiratie gerichte brainstormen. Dit zorgt voor het beste resultaat.

THE HOW OF THE RESEARCH

Dit zijn de manieren waarmee wij ons onderzoek gaan vormen.

Library

“Standing on the shoulds of giants.”

Library research is done to explore what is already done and what guidelines and theories exist that could help you further your design. Since the advent of the internet library research is also called desk research.

Field

“Understand your users.”

Field research is done to explore the application context. You apply a field strategy to get to know your end users, their needs, desires and limitations as organizational and physical contexts in which they will use your product.

Lab

“To measure is to know.”

Lab research is done to test parts or concepts of your product, of the final product. You use lab research to learn if things work out the way you intended them, or to test different scenarios.

Showroom

“Know & show your contribution.”

Showroom research is done to test your ideas in relation to existing work. Showing your prototype to experts can be a form of showroom research or spelling out how your product is different from the competition. Also testing your product to general guidelines is a form of showroom research.

Workshop

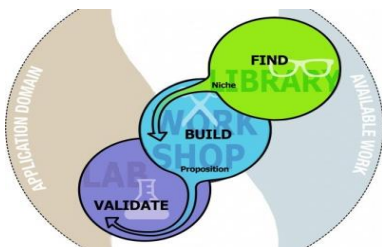
“Seek variation and improvement!”

Workshop research is done to explore opportunities. Prototyping, designing and co-creation activities are all ways to gain insights in what is possible and how things could work.








Dit is direct van <https://ictresearchmethods.nl/dot-framework/> (Vogel, n.d.)^[1]

TRIANGULATION

Een van de meest belangrijke onderdelen omdat het de verschillen methodes laat overlappen zodat er geen zwaktes meer in het onderzoek zitten.



https://ictresearchmethods.nl/_app/immutable/assets/400px-DesignPatternExampleForMixedMethods-bcf14b0b.jpg

Deelvraag/Onderwerp	Activiteit	Strategie	Methode	Uitkomst
Wat is een API key?	<ul style="list-style-type: none"> Onderzoek naar wat een API key is. Zijn er voorbeelden? 		<ul style="list-style-type: none"> Literature study available product analyses. 	<ul style="list-style-type: none"> Duidelijke uitleg van wat een API key is. Lijst met voorbeelden waar API keys gebruikt worden
Hoe is een API key ontstaan?	<ul style="list-style-type: none"> Onderzoek naar hoe een API key is ontstaan. 		<ul style="list-style-type: none"> Literature study 	<ul style="list-style-type: none"> Duidelijke uitleg van hoe de API key is ontstaan
Wanneer gebruik je een API key?	<ul style="list-style-type: none"> Onderzoek naar wanneer je een API key dient te gebruiken 		<ul style="list-style-type: none"> Literature study 	<ul style="list-style-type: none"> Duidelijke uitleg over wanneer er een API key gebruikt moet worden.
Wat is de beste manier om een API key te maken en te controleren?	<ul style="list-style-type: none"> Onderzoek naar beste manieren om API keys te maken. Prototype bouwen om goed te begrijpen hoe het werkt. Tests uitvoeren om te kijken of de key ook daadwerkelijk werkt 	  	<ul style="list-style-type: none"> Literature study Best good and bad practices Prototyping Security test 	<ul style="list-style-type: none"> Duidelijke uitleg over hoe er het beste een API key gemaakt kan worden. Prototype Testresultaten
Hoe sla je een API key op?	<ul style="list-style-type: none"> Onderzoek doen naar hoe je een API veilig op kan slaan 		<ul style="list-style-type: none"> Literature study Best good and bad practices 	<ul style="list-style-type: none"> Duidelijke uitleg over hoe je een API key veilig kan opslaan.

(Vogel, n.d.)^[1]

WAT IS EEN API KEY?

Een API-sleutel is een unieke identificatiecode die wordt gebruikt om een gebruiker, ontwikkelaar of roepend programma te authentifieren en autoriseren tot een API. Het fungeert als een geheim token dat alleen bekend is bij de client en de server. De API-sleutel wordt vaak verzonden in de querystring wanneer API-oproepen worden gemaakt en kan worden geassocieerd met specifieke servers waarop de aanroepende applicatie is geïmplementeerd (*What Is an API Key? | API Key Definition | Fortinet*, n.d.)^[2]. Hoewel ze vaak worden gebruikt om een project bij de API te authentifieren en te autoriseren, identificeren ze meestal geen menselijke gebruiker, maar eerder de aanroepende applicatie zelf (Wikipedia contributors, n.d.)^[3].

API-sleutels zijn meestal een unieke alfanumerieke string die in de API-oproep wordt opgenomen voor validatiedoeleinden. Ze spelen een cruciale rol bij het bijhouden van het gebruik van de API, het identificeren van ongeldige of kwaadaardige verzoeken en het beheren van de toegang tot de API. Elke API-sleutel wordt uitgegeven door een API-provider aan een geregistreerde consument en wordt bij elke aanvraag opgenomen, waarbij de API-server de API-sleutel controleert om de identiteit van de consument te bevestigen voordat de gevraagde gegevens worden geretourneerd (Team, n.d.)^[4].

Enkele voorbeelden:

1. **Okta API:** (dit is de service die wij gebruiken om onze applicatie te beveiligen door gebruik te maken van Auth0) Bij het aanmaken van een gebruikersaccount via Okta's API moet je jouw API-sleutel opnemen in het verzoek om het te laten slagen. De API-sleutel wordt toegevoegd in een Authorization HTTP request header met het formaat **Authorization: SSWS {jouw_api_sleutel}**. ^[6] (Okta, Edwards, n.d.)
2. **OpenWeather API:** Binnen onze applicatie denken wij eraan om een weather API toe te voegen aan onze applicatie. Een voorbeeld van hoe de weather API de key gebruikt: Nadat je je hebt aangemeld en een API-sleutel hebt ontvangen, moet deze bij elk verzoek worden meegestuurd. Je kunt het testen door een URL in je webbrowser te bezoeken, zoals http://api.openweathermap.org/data/2.5/weather?q=London,uk&APPID={jouw_api_sleutel} ^[7] (OpenWeatherMap.org, n.d.).
3. Een voorbeeld hoe een {jouw_api_key} eruit kan zien is bijvoorbeeld dit: `zaCElgL.0imfnc8mVLWwsAawjYr4Rx-Af50DDqtlx.`



HOE IS EEN API KEY ONTSTAAN?

De geschiedenis van API-Keys staan hand in hand met de ontwikkeling van web-API's. De moderne web-API's begonnen vorm te krijgen in de vroege jaren 2000, voortgezet door werk op het gebied van REST-API's door ROY Fielding en de commerciële competitie tussen eBay Amazon en Salesforce.

Rond 2004 begon het API-landschap te veranderen met de opkomst van sociale mediaplatforms zoals Facebook en Twitter en diensten zoals Flickr en Delicious, die API's ontwikkelden niet alleen voor commercieel gewin maar ook voor het delen van informatie en betrekken van gebruikers

De introductie van Amazon Web Services (AWS), zoals S3 en EC2, toonde het potentieel van API's in infrastructuur en clouddiensten, en vormde verder het moderne API-landschap

Met de lancering van de iPhone in 2007 en het Android-platform van Google, breidden API's uit naar mobiele toepassingen, waardoor de mogelijkheden van API's werden vergroot en een mobiel en onderling verbonden API-gedreven wereld werd gecreëerd.

Tegenwoordig zijn API's niet alleen voor web- en mobiele apps; ze zijn fundamenteel voor het "Internet van Alles", waarbij apparaten en diensten in verschillende industrieën worden verbonden en de weg wordt vrijgemaakt voor voortdurende digitale transformatie¹. (Lane, 2023b)[⁸]

De API-sleutels zijn dus in plaats gekomen omdat tijdens deze groei van de API de veiligheid er moet zijn om ongeautoriseerde toegang tot gegevens en diensten te voorkomen. Met de opkomst van web- en mobiele applicaties, evenals het Internet der Dingen, werd het steeds belangrijker om te controleren wie toegang heeft tot welke informatie. API-sleutels dienen als een eenvoudige maar effectieve manier om deze toegang te reguleren, door te verzekeren dat alleen geautoriseerde applicaties en ontwikkelaars toegang krijgen tot de API en de achterliggende gegevens. Dit helpt bij het handhaven van de integriteit, vertrouwelijkheid en beschikbaarheid van de API-gerelateerde diensten zoals onze API van TravelXPToday.

WANNEER GEBRUIK JE EEN API KEY?

Het gebruik van een API key is afhankelijk van de API. Dit ligt aan de eisen die de eigenaar van de API instelt. Een API kan bijvoorbeeld sommige methodes beperken door een API key te vereisen. Ook kun je instellen dat voor alle methodes een API key nodig is. Het is gebruikelijk om een API key te vereisen als je anoniem verkeer wilt blokkeren (*Why and when to use API keys*, z.d.) en ongeautoriseerde toegang tot de API wil voorkomen(Lane, 2023b)[⁸].

API keys worden gebruikt om het project te identificeren dat een API aanroept en om het gebruik van de API te volgen. Hieronder staan enkele voor en nadelen voor het gebruik van API keys. (Oltwater, 2022)

Voordelen:

- Eenvoudig: Het toevoegen van een API key aan een API request is vrij eenvoudig.
- Beperking van toegang: API keys worden gebruikt om de toegang tot de API te beperken en te controleren.
- Identificatie van de applicatie of het project: API keys identificeren de applicatie of het project dat een API aanroept.
- Bijhouden van statistieken: API keys kunnen worden gebruikt om het gebruik van de API te volgen en statistieken bij te houden.

Nadelen:

- Moeilijke identificatie: API keys identificeren alleen de applicatie en niet de gebruiker van de toepassing.
- Moeilijk om de key geheim te houden: API keys zijn niet zo veilig als authenticatietokens en kunnen gemakkelijk worden gestolen.

WAT IS DE BESTE MANIER OM EEN API KEY TE MAKEN EN TE CONTROLEREN?

MAKEN

Een klein overzichtje van de manieren waarop je een API-Key kan genereren met voor en na delen.

Random Bytes:

- **Voordelen:** Zeer willekeurig en veilig tegen pogingen tot raden.
- **Nadelen:** Kan niet direct worden gekoppeld aan gebruikersidentificatie zonder een extra opzoeksysteem.

UUID (Universally Unique Identifier):

- Voordelen: Makkelijk te genereren en verzekert uniciteit over verschillende systemen.
- Nadelen: Niet inherent veilig; afhankelijk van de implementatie kunnen ze voorspelbaar zijn.

Gebruik een CSPRNG (Cryptographically secure pseudorandom number generator): Er zijn meerdere betrouwbare libraries in bijna alle programmeer talen, Voorbeelden zijn `crypto.randomBytes` in Node.js, `secrets` in Python, of `RNGCryptoServiceProvider` in .NET. (Wikipedia contributors CSPRNG, z.d.) [¹¹] (Abaakouk, z.d.) [¹³]

- Voordelen: Bieden sterke en veilige willekeurige sleutelgeneratie.
- Nadelen: Vereisen extra afhankelijkheden en een goede kennis van cryptografische principes om veilig te implementeren.

Hashing:

- Voordelen: Goed voor het veilig opslaan van sleutels; de originele sleutel kan niet worden teruggewonnen uit de hash, wat een beveiligingstaak toevoegt.
- Nadelen: Als de Hashing niet correct wordt toegepast (bijvoorbeeld met een zwakke hashfunctie), kan het vatbaar zijn voor aanvallen.

Het meest belangrijke om aan te denken bij het genereren van een API-Key is: dat ze uniek zijn, willekeurig zijn en onvoorspelbaar. Dit betekent dus dat de API-Key gemaakt moet worden door de API zelf en niet de traveler op onze applicatie dit voorkomt dat de privileges verschillend blijven en dat alle Keys voldoen aan onze standaarden. (freeCodeCamp.org, z.d.) [¹²]

CONTROLEREN

Hier is een voorbeeld hoe wij een API-Key genereren using python secrets using secrets.token_urlsafe(32)

```
```python
import secrets

import hashlib

def generate_apikey(user_info):

 user_string = str(user_info)

 random_api_key = secrets.token_urlsafe(32)

 hashed_api_key = hashlib.sha512(random_api_key.encode('utf-8')).hexdigest()

 hashed_user = hashlib.sha512(user_string.encode('utf-8')).hexdigest()

 api_key = f"{hashed_api_key}-{hashed_user}"
```

```
 return api_key

print(generate_apikey("Test.test@gmail.com"))

def extract_api_key_and_user_info(api_key):

 api_key = api_key.split("-")

 random_api_key = api_key[0]

 hashed_user = api_key[1]

 return random_api_key, hashed_user

print(extract_api_key_and_user_info(generate_apikey("Test.test@gmail.com")))

'''
```

We gebruiken hier een hash algoritme genaamd SHA512. (Wikipedia contributors sha-2, z.d.) [<sup>16</sup>]

Als je meer wilt weten is hier de wiskunde achter de SHA512 algoritme : <https://en.wikipedia.org/wiki/SHA-2>

---

## HOE SLA JE EEN API KEY OP?

Om te controleren of de meegegeven API key correct is, dient deze key ook ergens te worden opgeslagen. Wanneer er een API request wordt uitgevoerd controleert het programma of de opgeslagen key overeenkomt met de meegegeven key in de API request. Er zijn verschillende manieren om API keys op te slaan. Het opslaan van de API key dient veilig te gebeuren. Dit is erg belangrijk. Hieronder staan enkele voorbeelden waar een API key veilig kan worden opgeslagen. (*Het beschermen van jouw Google API-key - Tijdvooreensite, z.d.*)<sup>[14]</sup>

- Gebruik een beveiligde database: Sla de API key op in een beveiligde database die alleen toegankelijk is voor geautoriseerde gebruikers. Deze autorisatie vereist meestal een gebruikersnaam en wachtwoord.
- Gebruik een beveiligde bestandsopslag: Sla de API key op in een beveiligd bestand dat alleen toegankelijk is voor geautoriseerde gebruikers. Let hierbij wel op dat dit bestand goed is beveiligd en niet online wordt gezet op bijvoorbeeld Github.
- Gebruik een beveiligde omgevingsvariabele: Sla de API key op als een omgevingsvariabele, bijvoorbeeld in Visual Studio secrets, die alleen toegankelijk is voor geautoriseerde gebruikers.
- Gebruik een beveiligde key management service: Sommige cloudproviders bieden key management services aan die kunnen worden gebruikt om API keys op te slaan en te beheren.
- Beperk de toegang tot de API key: Geef de API key alleen aan personen of bedrijven die je vertrouwt en beperk de toegang tot de API key tot alleen de noodzakelijke personen of applicaties.
- Vernieuw de API key regelmatig: Vernieuw de API key regelmatig om de beveiliging te verbeteren en ongeautoriseerde toegang te voorkomen.

Het is belangrijk om de API key op een veilige manier op te slaan om ongeautoriseerde toegang tot de API te voorkomen. Bijbehorend bij de bovenstaande opties, wordt het sterk aangeraden om je API key te hashen. Ondanks dat een API key niet makkelijk te raden is, is het belangrijk om je API key te hashen. Dit biedt extra beveiliging voor situaties die je niet zelf in de hand hebt. Denk hierbij aan bijvoorbeeld een datalek. (Oltwater, 2022b)<sup>[10]</sup>

## BRONVERMELDING

- [^1]: Vogel, J. (z.d.). ICT Research Methods — Methods Pack for Research in ICT. ICT Research Methods. Geraadpleegd op 8 november 2023, van <https://ictresearchmethods.nl/dot-framework/>
- [^2]: What Is an API Key? | API Key Definition | Fortinet. (z.d.). Fortinet. Geraadpleegd op 8 november 2023, van <https://www.fortinet.com/resources/cyberglossary/api-key>
- [^3]: Wikipedia contributors. (z.d.). *API key*. Wikipedia. Geraadpleegd op 8 november 2023, van [https://en.wikipedia.org/wiki/API\\_key](https://en.wikipedia.org/wiki/API_key).
- [^4]: Team, P. (z.d.). *What is an API key?* Postman Blog. Geraadpleegd op 8 november 2023, van <https://blog.postman.com/what-is-an-api-key/>
- [^5]: *Best practices for REST API security: Authentication and authorization - Stack Overflow*. (z.d.). Geraadpleegd op 8 november 2023, van <https://stackoverflow.blog/2021/10/06/best-practices-for-authentication-and-authorization-for-rest-apis/>
- [^6]: Edwards, P. (z.d.). *API Key Best Practices and Examples*. Okta Developer. Geraadpleegd op 8 november 2023, van <https://developer.okta.com/blog/2021/02/03/api-key-best-practices-and-examples>
- [^7]: OpenWeatherMap.org. (z.d.). *Weather API - OpenWeatherMap*. Openweather. Geraadpleegd op 8 november 2023, van <https://openweathermap.org/api>
- [^8]: Lane, K. (2023, 23 mei). *Intro to APIs: History of APIs*. Postman Blog. Geraadpleegd op 8 november 2023, van <https://blog.postman.com/intro-to-apis-history-of-apis/>
- [^9]: Why and when to use API keys. (z.d.). Google Cloud. Geraadpleegd op 8 november 2023, van <https://cloud.google.com/endpoints/docs/openapi/when-why-api-key>
- [^10]: Oltwater, S. (2022, 1 december). *Dé beste manier om security in te richten binnen je API gateway*. - eMagiz. eMagiz. <https://emagiz.com/blogs/de-beste-manier-om-security-in-te-richten-binnen-je-api-gateway/>
- [^11]: Wikipedia contributors CSPRNG. (z.d.). *Cryptographically secure pseudorandom number generator*. Wikipedia. Geraadpleegd op 8 november 2023, van [https://en.wikipedia.org/wiki/Cryptographically\\_secure\\_pseudorandom\\_number\\_generator](https://en.wikipedia.org/wiki/Cryptographically_secure_pseudorandom_number_generator)
- [^12]: freeCodeCamp.org. (z.d.). *Best practices for building secure API Keys*. Geraadpleegd op 8 november 2023, van <https://www.freecodecamp.org/news/best-practices-for-building-api-keys-97c26eabfea9/#:~:text=Af50DDqtlx>
- [^13]: Abaakouk, M. (z.d.). *On API Keys Best Practices*. The Mergify Blog. Geraadpleegd op 8 november 2023, van <https://blog.mergify.com/api-keys-best-practice/>
- [^14]: Het beschermen van jouw Google API-key - Tijdvooreensite. (z.d.). Geraadpleegd op 8 november 2023, van <https://www.tijdvooreensite.nl/blog/het-beschermen-van-jouw-google-api-key/>
- [^15]: Oltwater, S. (2022, 1 december). *Dé beste manier om security in te richten binnen je API gateway*. - eMagiz. eMagiz. <https://emagiz.com/blogs/de-beste-manier-om-security-in-te-richten-binnen-je-api-gateway/>
- [^16]: Wikipedia contributors sha-2. (z.d.). *SHA-2*. Wikipedia. Geraadpleegd op 8 november 2023, van <https://en.wikipedia.org/wiki/SHA-2>

<https://www.freecodecamp.org/news/best-practices-for-building-api-keys-97c26eabfea9/#:~:text=Since%20the%20API%20key%20itself,0imfnc8mVLWwsAawjYr4Rx-Af50DDqtlx%20.>  
<https://stackoverflow.blog/2021/10/06/best-practices-for-authentication-and-authorization-for-rest-apis/>

implementation

<https://blog.teclado.com/api-key-authentication-with-flask/>